REMARKS

Reconsideration of the present application is respectfully requested. Claims 1-45 were pending. Claims 1 and 41 have been amended without introducing any new matter. No claims have been added or cancelled. Thus, claims 1-45 remain pending.

Claims 1-12, 15, 17-20, 21-32, 35, 37-41, 43, and 44 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Publication No. 2002/0129060 of Rollins et al. (hereinafter "Rollins") in view of U.S. Patent No. 6,675,354 of Claussen et al. (hereinafter "Claussen"). The Applicant respectfully maintains the right to swear behind the cited references at a later date. The Applicant respectfully disagrees with the rejection because Rollins and Claussen, alone or in combination, fail to teach or suggest each and every element of the invention claimed by the Applicant in claims 1-12, 15, 17-20, 21-32, 35, 37-41, 43, and 44.

Rollins describes a system and method for generating multiple customizable interfaces for XML documents (Rollins, Abstract). A component generation engine uses XML schema, which defines the structure of an XML document, and optional user preferences to define multiple components (Rollins, page 2, paragraph 0031; page 3, paragraphs 36-37). Then, a code-generator generates "a set of Java classes designed to mediate communication between the user and the synchronized tree manager" (Rollins, page 3, paragraph 0038). However, the generated classes are only described as java classes which minimally implement Rollins's mediator interface.

Claussen describes processing and handling custom tags in a document (Claussen, Column 3, lines 14-29). The custom tags may be processed regardless of the type of tag that is encountered (Claussen, Column 30, lines 30-42). To avoid errors

in processing the various tags, a tag preprocessing technique is utilized to classify the

tag according to a tag library (Claussen, Column 6, lines 8-32). However, Claussen

merely describes routines for handling various custom tags.

With respect to claim 1, the Applicant claims:

In a computer system, an improved method for developing a Web
application, the method comprising:
providing a Web application framework, said framework including
an abstract command tag that predefines at least some generic Web
application activities;
specifying at least one custom action that is desired to be
performed by a Web application;
creating an object oriented programming (OOPL) class that extends
the abstract command tag for providing execution logic for said at least
one custom action, in addition to pre-existing logic that supports said at
least some generic Web application activities, thereby creating a
corresponding customized command tag that is capable of being
embedded within a Web page;
embedding the customized command tag in a Web page of the
Web application; and
upon execution of the Web application including an embedded
customized command tag in a Web page, invoking the customized
command tag for conditionally executing said specified at least one
custom action based on run-time conditions.

The Applicant respectfully submits that Rollins and Claussen, alone or in combination,

fail to teach or suggest "creating an object oriented programming (OOPL) class that

extends the abstract command tag ..., thereby creating a corresponding customized

command tag [and] invoking the customized command tag." Claussen merely

describes handling custom tags but fails to describe extending an abstract command

tag thereby creating a corresponding customized command tag. Although Rollins

describes generating Java classes which mediate communication between a user and

an XML document, the classes are not described as being classes for tags. Further, at

no point does Rollins refer to, or suggest, the use and extension of an abstract command tag.

The Examiner argued that Rollins discloses a "set of java classes is the results from the customized tags as described by the user (page 3, [0038])" (Office Action, page 3). Further, "Rollins discloses XML application development, which is done on the Web environment and used by the World Wide Web (page 2 [0015-0016])" while "Claussen disclosures processing the custom tag in a document object model (DOM) representation that is an internal XML document data structure representation and basically a tree of all nodes in an XML file (col. 3, lines 14-52)" (Office Action, page 4). The Examiner asserted that the passages disclose the limitations of "providing a Web application development framework, said framework including an abstract command tag that predefines at least some generic Web application activities [and] an object oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action," as claimed by the Applicants in claim 1.

The Applicants respectfully disagree. Rollins discloses java classes which are isolated from a provided library of user interface classes, so that the subgroup of classes is capable of performing a customized rule in the Rollins interface (Rollins, page 3, paragraph [0038]; page 2, paragraph [0015]). Thus Rollins merely describes a subgroup of predefined classes. The abstract command tag, however, is a java element that is extended with new code, but not a regrouping of existing code, to provide for custom functionality of the abstract tag. Because extending a class is not

the same as creating a subgroup of existing classes, Rollins fails to describe or suggest an abstract command tag or extending the abstract command tag.

Furthermore, Claussen merely describes a "processing the custom tag in a document object model (DOM) representation that is an internal XML document data structure representation and basically a tree of all nodes in an XML file" (Office Action, page 4 *citing* Claussen, col. 3, lines 14-52). While Claussen describes "processing" custom XML tags in a DOM model, the processing performed by Claussen directs the custom tag to the appropriate tag handler. Processing the custom tag by directing a tag to the appropriate handler, however, does not describe <u>providing</u> an abstract tag and extending the abstract tag to perform a custom action.

Therefore, neither Rollins nor Claussen, alone or in combination, teaches or suggest "providing a Web application development framework, said framework including an abstract command tag that predefines at least some generic Web application activities [and] an object oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action," as claimed in Claim 1. Thus, claims 2-12, 15, and 17-20, which depend from claim 1, are also not rendered obvious by Rollins in view of Claussen. Therefore, the Applicants respectfully request withdrawal of the rejection of claims 1-12, 15, and 17-20 under 35 U.S.C. § 103.

Similarly, claim 21 recites:

> A Web application framework comprising:
> specification of an abstract command tag that predefines at least some generic Web application activities;
> a programming environment for:
> (i) specifying at least one custom action that is desired to be performed by a Web application under development, by supporting

creation of a object-oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action, thereby creating a corresponding customized command tag that is capable of being embedded within a Web page, wherein said customized command tag includes the ability to conditionally execute said specified at least one custom action based on run-time conditions; and

(ii) enabling embedding the customized command tag in a Web page of the Web application;

wherein execution of the Web application includes invocation of the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions.

As discussed above with respect to claim 1, neither Rollins nor Claussen, alone or in combination teach or suggest the use and extension of an abstract command tag. Rollins merely discusses creating Java classes for facilitating communication between a user and an XML document, while Claussen simply handles custom tags. Therefore, since neither reference, alone or in combination teaches or suggests the use and extension of an abstract command tag, claim 21 is not rendered obvious by Rollins in view of Claussen. Furthermore, claims 22-32, 35, 37-40 depend on claim 21, and include additional features and limitations. Thus, claims 22-32, 35, 37-40 are also not obvious under Rollins in view of Claussen.

Claim 41 recites:

An improved method for Web application development, the method comprising:
providing a Web-based application development framework built from a set of object-oriented programming language (OOPL) classes, which extends an abstract command tag, said framework providing:
a non-programmatic tag framework that implements the functionality of the application framework when executing within a Web page using dynamic scripting capability;
tag-based Web application objects controlling program flow, executing user commands, representing application business objects, and constructing output;

a non-programmatic tag framework that accesses data for logical business objects and allows page designers to specify an action to be performed;

enabling the embedding of the tag-based Web application objects in a Web page of a Web application; and

execution of the Web application including invoking the tag-based Web application objects.

As discussed above, with respect to claims 1 and 21, Rollins and Claussen, alone or in combination, fail to teach or suggest an extended abstract command tag. Therefore, Rollins and Claussen fail to render claim 41 obvious. Furthermore, since claims 43 and 44 contain additional features and limitations to those claimed in claim 41, claims 43 and 44 are also not rendered obvious by Rollins in view of Claussen. The Applicant respectfully requests withdrawal of the rejections.

The Examiner rejected claims 13, 14, 16, 33, 34, 36, 42, and 45 under 35 U.S.C. § 103(a) as being unpatentable over Rollins and Claussen, in view of U.S. Patent No. 6,760,748 of Hakim (hereinafter "Hakim"). As discussed above with respect to claims 1, 21, and 41, from which claims 13, 14, 16, 33, 34, 36, 42, and 45 depend, Rollins and Claussen fail to describe each and every limitation claimed by the Applicant. Because Hakim merely describes an electronic classroom in which data is transmitted from a teacher's terminal to student terminals (Hakim, Column 3, lines 15-21; Figure 2), Hakim also fails to teach or suggest the limitations discussed above with respect to claims 1, 21, and 41. Thus, Rollins, Claussen, and Hakim, alone or in combination, fail to render claims 1, 21, and 41, and thus dependent claims 13, 14, 16, 33, 34, 36, 42, and 45, obvious.

In view of the foregoing amendments and remarks, Applicants respectfully submit that all pending claims are in condition for allowance, and such action is respectfully

requested. If a telephone interview would expedite the prosecution of this application, the Examiner is invited to contact Judith Szepesi at (408) 720-8300.
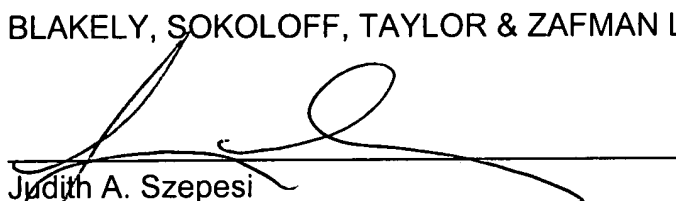
If there are any additional charges/credits, please charge/credit our deposit account no. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 6/20/06

Judith A. Szepesi
Reg. No. 39,393

Customer No. 08791
12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025
(408) 720-8300